In the Claims:

1.   (Currently amended) A computer-implemented method for collecting information relating to execution of an application, the method comprising:

determining a set of probe locations in the application, wherein the set of probe locations comprises a probe location pair that produces redundant information at which collecting data relating to the execution of the application would produce non-redundant information;

eliminating one member of the probe location pairs of probe locations that would produce redundant information, wherein the probe location pair comprises one of the following pairs: a call to function probe location in a calling function and an enter function probe location in a called function, and a return from function probe location in the calling function and an exit function probe location in the called function eliminating occurs at a time other than when the application is being compiled; and

inserting probes only at the remaining determined probe locations in the application such that data collected relating to the execution of the application produces non-redundant information.

Claims 2-3   (Cancelled)

4.   (Original)   The method of claim 1, further comprising:

identifying a first location within the application at which a function call directs execution of the application to a second location outside of a current module; and

inserting a first probe before the identified first location and a second probe after the identified first location.

5.   (Original)   The method of claim 4, wherein the first probe is configured to collect an address of a first function in which the identified first location is located, an address of a second function in which the second location is located, a first stack pointer, and a first time

indicator, and the second probe is configured to collect the address of the second function, a second stack pointer, and a second time indicator.

6.   (Currently amended) The method of claim 1, further comprising:

identifying a first location within the ~~a~~ calling function at which execution of the application is directed to the ~~a~~ called function having an exit point at which execution of the application is directed to a second location outside of the calling function; and

inserting a first probe before the first location and a second probe after the second location.

7.   (Original)   The method of claim 6, wherein the first probe is configured to collect an address of the calling function, an address of the called function, a first stack pointer, and a first time indicator, and the second probe is configured to collect the address of the called function, a second stack pointer, and a second time indicator.

8.   (Original)   The method of claim 1, further comprising:

identifying a block of code to which execution of the application is directed upon occurrence of an error; and

inserting a first probe at a beginning of the identified block of code and a second probe at an end of the identified block of code.

9.   (Original)   The method of claim 8, wherein the first probe is configured to collect an address of the block of code, a first stack pointer, and a first time indicator, and the second probe is configured to collect the address of the block of code, a second stack pointer, and a second time indicator.

10.   (Original)   The method of claim 1, further comprising using the inserted probes to collect the information relating to the execution of the application.

11.   (Original)   The method of claim 10, further comprising analyzing the collected information.

12.   (Currently amended)   A computer-implemented method for collecting information relating to execution of an application, the method comprising:

determining entry and exit points of a called ~~plurality of~~ functions that is called from a calling function in ~~constituting at least a portion of~~ the application;

determining a set of probe locations in the application, wherein the set of probe locations comprises probe location pairs that produce redundant information ~~at which collecting data relating to the execution of the application would produce non-redundant information~~;

eliminating one member of each probe location pairs ~~of probe locations that would produce redundant information~~, wherein each probe location pair comprises one of the following pairs: a call to function probe location in the calling function and an enter function probe location at the entry point in the called function, and a return from function probe location in the called function and an exit function probe location at the exit point in the called function ~~the eliminating occurs at a time other than when the application is being compiled, and wherein the set of probe locations includes at least the entry and exit points of the functions~~;

inserting probes ~~only~~ at the remaining ~~determined~~ probe locations in the application;

collecting ~~the~~ non-redundant information relating to the execution of the application using the inserted probes; and

analyzing the collected information.

13.   (Original)   The method of claim 12, further comprising:

identifying a first location within the application at which a function call directs execution of the application to a second location outside of a current module; and

inserting a first probe before the identified first location and a second probe after the identified first location,

wherein the first probe is configured to collect an address of a first function in which the identified first location is located, an address of a second function in which the second

location is located, a first stack pointer, and a first time indicator, and the second probe is configured to collect the address of the second function, a second stack pointer, and a second time indicator.

14.   (Original)   The method of claim 12, further comprising:

identifying a first location within a calling function at which execution of the application is directed to a called function having an exit point at which execution of the application is directed to a second location outside of the calling function; and

inserting a first probe before the first location and a second probe after the second location,

wherein the first probe is configured to collect an address of the calling function, an address of the called function, a first stack pointer, and a first time indicator, and the second probe is configured to collect the address of the called function, a second stack pointer, and a second time indicator.

15.   (Original)   The method of claim 12, further comprising:

identifying a block of code to which execution of the application is directed upon occurrence of an error; and

inserting a first probe at a beginning of the identified block of code and a second probe at an end of the identified block of code,

wherein the first probe is configured to collect an address of the block of code, a first stack pointer, and a first time indicator, and the second probe is configured to collect the address of the block of code, a second stack pointer, and a second time indicator.

16.   (Currently amended) A computer-readable medium having an application including computer-executable instructions for:

determining a set of probe locations in the application, wherein the set of probe locations includes a probe location pair that produces redundant information at which collecting data relating to the execution of the application would produce non-redundant information;

eliminating <u>one member of the probe location</u> pairs ~~of probe locations that would~~ ~~produce redundant information~~, wherein the <u>probe location pair comprises one of the following</u> <u>pairs: a call to function probe location in a calling function and an enter function probe location</u> <u>in a called function, and a return from function probe location in the calling function and an exit</u> <u>function probe location in the called function</u> ~~eliminating occurs at a time other than when the~~ ~~application is being compiled~~; and

inserting probes ~~only~~ at the <u>remaining</u> ~~determined~~ probe locations in the application <u>such that data collected relating to the execution of the application produces</u> <u>non-redundant information</u>.

Claims 17-18   (Cancelled)

19.   (Original)   The computer-readable medium of claim 16, having further computer-executable instructions for:

identifying a first location within the application at which a function call directs execution of the application to a second location outside of a current module; and

inserting a first probe before the identified first location and a second probe after the identified first location.

20.   (Original)   The computer-readable medium of claim 19, wherein the first probe is configured to collect an address of a first function in which the identified first location is located, an address of a second function in which the second location is located, a first stack pointer, and a first time indicator, and the second probe is configured to collect the address of the second function, a second stack pointer, and a second time indicator.

21.   (Currently amended) The computer-readable medium of claim 16, having further computer-executable instructions for:

identifying a first location within <u>the</u> ~~a~~ calling function at which execution of the application is directed to <u>the</u> ~~a~~ called function having an exit point at which execution of the application is directed to a second location outside of the calling function; and

inserting a first probe before the first location and a second probe after the second location.

22. (Original) The computer-readable medium of claim 21, wherein the first probe is configured to collect an address of the calling function, an address of the called function, a first stack pointer, and a first time indicator, and the second probe is configured to collect the address of the called function, a second stack pointer, and a second time indicator.

23. (Original) The computer-readable medium of claim 16, having further computer-executable instructions for:

identifying a block of code to which execution of the application is directed upon occurrence of an error; and

inserting a first probe at a beginning of the identified block of code and a second probe at an end of the identified block of code.

24. (Original) The computer-readable medium of claim 23, wherein the first probe is configured to collect an address of the block of code, a first stack pointer, and a first time indicator, and the second probe is configured to collect the address of the block of code, a second stack pointer, and a second time indicator.

25. (Original) The computer-readable medium of claim 16, having further computer-executable instructions for using the inserted probes to collect the information relating to the execution of the application.

26. (Original) The computer-readable medium of claim 25, having further computer-executable instructions for analyzing the collected information.

27. (Currently amended) A computer-readable medium having an application including computer-executable instructions for:

determining entry and exit points of a <u>called</u> ~~plurality of~~ functions <u>that is called</u> <u>from a calling function in</u> ~~constituting at least a portion of~~ the application;

determining a set of probe locations in the application<u>, wherein the set of probe</u> <u>locations comprises probe location pairs that produce redundant information</u> ~~at which collecting~~ ~~data relating to the execution of the application would produce non-redundant information~~;

eliminating <u>one member of each probe location</u> pairs ~~of probe locations that~~ ~~would produce redundant information~~, wherein <u>each probe location pair comprises one of the</u> <u>following pairs: a call to function probe location in the calling function and an enter function</u> <u>probe location at the entry point in the called function, and a return from function probe location</u> <u>in the called function and an exit function probe location at the exit point in the called function</u> ~~the eliminating occurs at a time other than when the application is being compiled, and wherein~~ ~~the set of probe locations includes at least the entry and exit points of the functions~~;

inserting probes ~~only~~ at the <u>remaining</u> ~~determined~~ probe locations in the application;

collecting ~~the~~ non-redundant information relating to the execution of the application using the inserted probes; and

analyzing the collected information.

28.   (Original)   The computer-readable medium of claim 27, having further computer-executable instructions for:

identifying a first location within the application at which a function call directs execution of the application to a second location outside of a current module; and

inserting a first probe before the identified first location and a second probe after the identified first location,

wherein the first probe is configured to collect an address of a first function in which the identified first location is located, an address of a second function in which the second location is located, a first stack pointer, and a first time indicator, and the second probe is configured to collect the address of the second function, a second stack pointer, and a second time indicator.

29.   (Currently amended) The computer-readable medium of claim 27, having further computer-executable instructions for:

identifying a first location within the ~~a~~ calling function at which execution of the application is directed to the ~~a~~ called function having an exit point at which execution of the application is directed to a second location outside of the calling function; and

inserting a first probe before the first location and a second probe after the second location,

wherein the first probe is configured to collect an address of the calling function, an address of the called function, a first stack pointer, and a first time indicator, and the second probe is configured to collect the address of the called function, a second stack pointer, and a second time indicator.

30.   (Original)   The computer-readable medium of claim 27, having further computer-executable instructions for:

identifying a block of code to which execution of the application is directed upon occurrence of an error; and

inserting a first probe at a beginning of the identified block of code and a second probe at an end of the identified block of code,

wherein the first probe is configured to collect an address of the block of code, a first stack pointer, and a first time indicator, and the second probe is configured to collect the address of the block of code, a second stack pointer, and a second time indicator.

31.   (Currently amended) A computer arrangement configured to execute an application including computer-executable instructions for:

determining a set of probe locations in the application, wherein the set of probe locations includes a probe location pair that produces redundant information ~~at which collecting data relating to the execution of the application would produce non-redundant information~~;

eliminating one member of the probe location pairs ~~of probe locations that would produce redundant information~~, wherein the probe location pair comprises one of the following pairs: a call to function probe location in a calling function and an enter function probe location

in a called function, and a return from function probe location in the calling function and an exit function probe location in the called function eliminating occurs at a time other than when the application is being compiled; and

inserting probes only at the remaining determined probe locations in the application such that data collected relating to the execution of the application produces non-redundant information.

Claims 32-33 (Cancelled)

34. (Original) The computer arrangement of claim 31, further configured to execute computer-executable instructions for:

identifying a first location within the application at which a function call directs execution of the application to a second location outside of a current module; and

inserting a first probe before the identified first location and a second probe after the identified first location.

35. (Original) The computer arrangement of claim 34, wherein the first probe is configured to collect an address of a first function in which the identified first location is located, an address of a second function in which the second location is located, a first stack pointer, and a first time indicator, and the second probe is configured to collect the address of the second function, a second stack pointer, and a second time indicator.

36. (Currently amended) The computer arrangement of claim 31, further configured to execute computer-executable instructions for:

identifying a first location within the a calling function at which execution of the application is directed to the a called function having an exit point at which execution of the application is directed to a second location outside of the calling function; and

inserting a first probe before the first location and a second probe after the second location.

37.   (Original)   The computer arrangement of claim 36, wherein the first probe is configured to collect an address of the calling function, an address of the called function, a first stack pointer, and a first time indicator, and the second probe is configured to collect the address of the called function, a second stack pointer, and a second time indicator.

38.   (Original)   The computer arrangement of claim 31, further configured to execute computer-executable instructions for:

identifying a block of code to which execution of the application is directed upon occurrence of an error; and

inserting a first probe at a beginning of the identified block of code and a second probe at an end of the identified block of code.

39.   (Original)   The computer arrangement of claim 38, wherein the first probe is configured to collect an address of the block of code, a first stack pointer, and a first time indicator, and the second probe is configured to collect the address of the block of code, a second stack pointer, and a second time indicator.

40.   (Original)   The computer arrangement of claim 31, further configured to execute computer-executable instructions for using the inserted probes to collect the information relating to the execution of the application.

41.   (Original)   The computer arrangement of claim 40, further configured to execute computer-executable instructions for analyzing the collected information.

42.   (Currently amended) A computer arrangement configured to execute an application including computer-executable instructions for:

determining entry and exit points of a called ~~plurality of~~ functions that is called from a calling function in ~~constituting at least a portion of~~ the application;

determining a set of probe locations in the application, wherein the set of probe locations comprises probe location pairs that produce redundant information ~~at which collecting data relating to the execution of the application would produce non-redundant information~~;

eliminating one member of each probe location pairs ~~of probe locations that would produce redundant information, the set of probe locations including at least the entry and exit points of the functions~~, wherein each probe location pair comprises one of the following pairs: a call to function probe location in the calling function and an enter function probe location at the entry point in the called function, and a return from function probe location in the called function and an exit function probe location at the exit point in the called function ~~the eliminating occurs at a time other than when the application is being compiled~~;

inserting probes ~~only~~ at the remaining ~~determined~~ probe locations in the application;

collecting ~~the~~ non-redundant information relating to the execution of the application using the inserted probes; and

analyzing the collected information.

43.   (Original)   The computer arrangement of claim 42, further configured to execute computer-executable instructions for:

identifying a first location within the application at which a function call directs execution of the application to a second location outside of a current module; and

inserting a first probe before the identified first location and a second probe after the identified first location,

wherein the first probe is configured to collect an address of a first function in which the identified first location is located, an address of a second function in which the second location is located, a first stack pointer, and a first time indicator, and the second probe is configured to collect the address of the second function, a second stack pointer, and a second time indicator.

44.   (Currently amended)   The computer arrangement of claim 42, further configured to execute computer-executable instructions for:

identifying a first location within the a calling function at which execution of the application is directed to the a called function having an exit point at which execution of the application is directed to a second location outside of the calling function; and

inserting a first probe before the first location and a second probe after the second location,

wherein the first probe is configured to collect an address of the calling function, an address of the called function, a first stack pointer, and a first time indicator, and the second probe is configured to collect the address of the called function, a second stack pointer, and a second time indicator.

45.   (Original)   The computer arrangement of claim 42, further configured to execute computer-executable instructions for:

identifying a block of code to which execution of the application is directed upon occurrence of an error; and

inserting a first probe at a beginning of the identified block of code and a second probe at an end of the identified block of code,

wherein the first probe is configured to collect an address of the block of code, a first stack pointer, and a first time indicator, and the second probe is configured to collect the address of the block of code, a second stack pointer, and a second time indicator.

Claims 46-51 (Cancelled)